

Elmar Schömer  
Ann-Christin Wörl

## 6. Übungsblatt

**Abgabe:** Dienstag, der 05.12.2023, 14:00 Uhr

### Aufgabe 1: Laufzeiten

(8+7 Punkte)

Das folgende Python-Programm erstellt ein Feld  $A$  der Länge  $n$  mit den Einträgen  $0, 1, \dots, n-1$ , indem sukzessive Elemente mit Hilfe der Operation `append` am Feldende angefügt werden. Im Gegensatz dazu wird das Feld  $B$  so aufgebaut, dass die Einträge  $n-1, n-2, \dots, 1, 0$  in dieser Reihenfolge am Feldanfang mit Hilfe der Operation `insert(0, .)` eingefügt werden.

```
1  n = 10
2
3  A = []
4  for i in range(n):
5      A.append(i)
6
7  B = []
8  for i in range(n-1, -1, -1):
9      B.insert(0, i)
10
11 print(A == B)
```

1. Führen Sie Laufzeitmessungen für  $n = 10^k$  für  $k = 1, 2, \dots, 6$  durch und ermitteln Sie experimentell, wie schnell die beiden Verfahren zum Aufbau des Feldes  $A$  und des Feldes  $B$  sind. Wie lange würde es auf Ihrem Rechner dauern, das Feld  $B$  für  $n = 10^7$  aufzubauen?
2. Leiten Sie anhand einer theoretischen Überlegung das unterschiedliche Laufzeitverhalten her, und vergleichen Sie das theoretische Resultat mit der praktischen Laufzeitmessung.

### Aufgabe 2: Copy

(8+7 Punkte)

1. Erklären Sie das Verhalten des folgenden Python-Programmes.

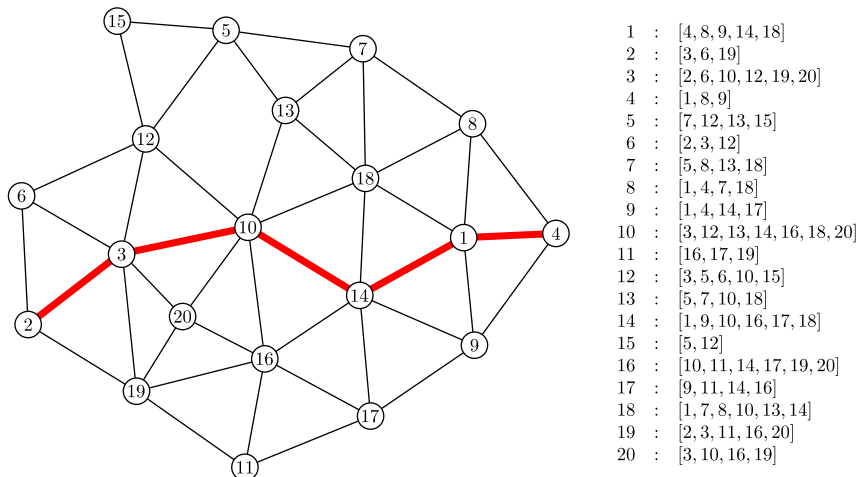
```
1  def copy(X):
2      Y = []
3      for x in X:
4          Y.append(x)
5      return(Y)
6
7  a = [1, 2, 3, 4]
8  b = a
9  c = copy(a)
10
11 a[3] = 42
12
13 print(a, b, c)
14
15 A = [[1, 2], [3, 4]]
16 B = A
17 C = copy(A)
18
19 A[1][1] = 42
20
21 print(A, B, C)
```

2. Schreiben Sie eine Funktion namens `deepcopy` zum Kopieren eines eindimensionalen Feldes, dessen Feldeinträge selbst eindimensionale Felder unterschiedlicher Länge sind (z.B. Adjazenzlisten eines Graphen). Die erzeugte "tiefe Kopie" soll die Eigenschaft besitzen, dass nach dem Funktionsaufruf  $C = \text{deepcopy}(A)$  Änderungen an  $A$  keine Auswirkungen auf  $C$  zur Folge haben.

**Aufgabe 3: Ungerichtete Graphen**

(5+5+5+5 Punkte)

Wir betrachten einen ungerichteten Graphen  $G = (V, E)$  und seine Darstellung durch Adjazenzlisten. Eine ungerichtete Kante zwischen zwei Knoten  $u$  und  $v$  bedeutet, dass man von  $u$  nach  $v$  gelangen kann und ebenso von  $v$  nach  $u$ . Man kann eine ungerichtete Kante also durch zwei gerichtete Kanten in entgegengesetzte Richtungen ersetzen.



1. Ändern Sie das Beispielprogramm zur Breitensuche so ab, dass die Graphen aus `huepfburg?.txt` als ungerichtete Graphen eingelesen werden und bauen Sie dazu die Adjazenzlisten auf.
2. Wir bezeichnen mit  $d(u, v)$  die Länge eines kürzesten Weges von  $u$  nach  $v$ , also eines Weges der die kleinst mögliche Zahl von Kanten benutzt, um von  $u$  nach  $v$  zu gelangen. Ergänzen Sie die Breitensuche so, dass es auch möglich ist,  $d(s, v)$  für einen Startknoten  $s$  zu alle Zielknoten  $v \in V$  in Form eines Distanzfeldes `dist` zu berechnen:  $\text{dist}[v] = d(s, v)$ .
3. Berechnen Sie für alle Paare von Knoten eine Abstandstabelle in Form eines zweidimensionalen Feldes!
4. Unter dem Durchmesser eines ungerichteten Graphen versteht man die größte auftretende kürzeste Weglänge zwischen zwei Knoten des Graphen.

$$\text{Diameter}(G) = \max\{d(u, v) | u, v \in V\}$$

Bestimmen Sie jeweils den Durchmesser der Graphen `huepfburg?.txt`!

#### Aufgabe 4: Zauberschule (BWINF)

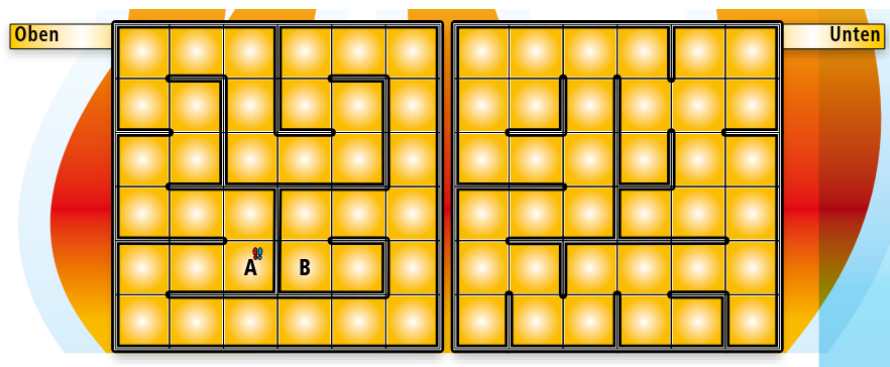
(20+10 Punkte)

##### In Anlehnung an Aufgabe 3 des 42. Bundeswettbewerbs Informatik: Zauberschule

Die Zauberschule Bugwarts hat zwei Stockwerke. Die Stockwerke liegen genau übereinander. Beide sind in Felder eingeteilt, und es gibt Wände zwischen einigen Feldern.

Zauberschüler Ron hat es immer eilig. Wenn er von einem Feld zu einem anderen gelangen will, soll das so schnell wie möglich gehen. Ron braucht 1 Sekunde, um auf dem gleichen Stockwerk von einem Feld zum nächsten zu gelangen. Leider hat Ron vergessen, wie er durch Wände gehen kann. Er kann aber von einem Stockwerk zum entsprechenden Feld des anderen Stockwerks gelangen; dazu braucht er 3 Sekunden.

Das Bild zeigt einen Plan der Zauberschule. Ron kann in 7 Sekunden von Feld A zu Feld B gelangen: In 3 Sekunden wechselt er ins untere Stockwerk, geht dort in 1 Sekunde ein Feld nach rechts und wechselt in weiteren 3 Sekunden zurück ins obere Stockwerk. Ein Weg nur durch das obere Stockwerk würde mindestens 13 Sekunden dauern.



1. Hilf Ron und schreibe ein Programm, das für jeweils zwei Felder A und B in der Zauberschule die Länge des kürzesten Weges zurückgibt, auf dem Ron so schnell wie möglich von A nach B gelangt. Das Programm soll dazu auch den Plan von Bugwarts – und jeden anderen Plan der gleichen Art, in dem nur die Wände anders stehen – einlesen können. So kann Ron das Programm auch Hermine zur Verfügung stellen, die in der Zweigstelle Fameglitch Nachhilfeunterricht gibt.
2. Visualisiere den kürzesten Weg von A nach B. Der Pfad selbst kann zum Beispiel mit <, >, ^ und v für die Richtungen und ! für den Wechsel beschrieben werden.

Wende dein Programm mindestens auf alle Beispiele `zauberschule?.txt` an. Die Dateien sind dabei wie folgt aufgebaut:

- In der ersten Zeile steht die Dimension  $n, m$  jedes Stockwerks. Die folgenden  $n$  Zeilen stellen das erste Stockwerk dar, gefolgt von einer Leerzeile und anschließend die  $n$  Zeilen des zweiten Stockwerks
- A steht für den Start, B für das Ziel
- Eine Wand wird durch # dargestellt, ein freies Feld durch .
- `zauberschule0.txt` entspricht dem oben beschriebenen Beispiel. Durch die Textdarstellung ist die Wand selbst ein Zeichen dick, sodass Ron 8 Sekunden benötigt, anstatt 7.

**Tipp:** Die Breitensuche ignoriert Kantengewichte und gibt die minimale Anzahl an Kanten zwischen zwei Knoten zurück. Um die längere Dauer des Stockwerkwechsels im Vergleich zum Wechsel innerhalb des Stockwerks zu berücksichtigen, können zusätzliche, "imaginäre" Knoten in die Adjazenzliste eingefügt werden.

### Aufgabe 5: Wortsalat

(5+10+5 Punkte)

Die Joannes Gutenberg - Universität Mainz (JGU) zählt mit mehr als 300.000 Studierenden aus 120 Nationen zu den größten und vielfältigsten Universitäten Deutschlands. Mit ihrer Uniäzessivität, ihren Hochschulen für Kunst und Musik sowie ihrem Fachbereich Transdisziplinäre, Sprach- und Kulturenwissenschaft in Gießen verient sie nahezu alle akademischen Disziplinen. In ihren mehr als 100 Instituten und Kliniken lehren und forschen rund 4.500 Wissenschaftlerinnen und Wissenschaftler, darunter 570 Professoren und Professorinnen. Mit 76 Fakultäten und zahlreichen Fakultätsabteilungen bietet sie 296 Studiengänge an. Als einzige deutsche Universität ihrer Größe beherbergt die JGU fast alle Institute auf einem zentralen Campus, auf dem auch fünf Fakultäten der Medizin und der Naturwissenschaften angesiedelt sind: das Max - Planck - Institut für Chemie, das Max - Planck - Institut für Polymerforschung, das Heinrich - Heine - Institut Mainz, das Institut für Biochemie und Mikrobiologie - Biophysik und das Institut für Molekulare Biologie. Der Campus der Universität Mainz liegt nur etwa einen Kilometer entfernt, die drei Mainzer Institute der Leibniz - Geisteswissenschaften - das Institut für Europäische Geschichte, das Leibniz - Institut für Wissenssoziologie und das Römisch - Germanische Zentralmuseum - liegen nur wenig weiter entfernt in der Innenstadt. Hinzu kommen zahlreiche Fachbereiche und Fakultäten, die zu der hohen Dynamik des Universitätsstandorts Mainz beitragen.

1. Schreiben Sie eine Funktion zum Lesen und Zerlegen eines Textes. Als Argument soll diese Funktion einen Dateinamen erhalten und ein Feld der Wörter in der gelesenen Reihenfolge zurückliefern. (Der Einfachheit halber sollen Satzzeichen einfach zu dem Wort gezählt werden, auf das sie folgen.)
2. Schreiben Sie eine Funktion, die einen String in einzelne Buchstaben zerlegt und diese in einem Feld ablegt. Dann sollen die Buchstaben zufällig permutiert und wieder zu einem String zusammengesetzt werden. Die Permutation soll so erfolgen, dass die ersten beiden und die letzten beiden Buchstaben an ihrer alten Position bleiben. Die Funktion erhält als Argument einen String und liefert als Rückgabewert ebenfalls einen String. Verwenden Sie außer `random` keine zusätzlichen Bibliotheken.
3. Nutzen Sie die beiden Funktionen, um ein Programm zu schreiben, das den Text `JGU.txt` in eine ähnlich (un)lesbare Form wie oben verwandelt.